퐁 음영법을 위한 3차원 그래픽 가속기의 구현

이 형[†] · 박윤옥[††] · 박종원[†††]

요 약

CAD/CAM, 3차원 모델링, 가상현실, 그리고 의학 영상의 처리 속도를 높이기 위한 3차원 가속기에 대한 많은 연구들이 진행 중이다. 본 논문에서는 3차원 그래픽 처리속도를 향상하기 위하여 SIMD처리기 구조의 3차원 가속기를 제안하며, 기존의 퐁 음영법을 제안된 구조에 맞게 병렬화하고 수행함으로써 직접적인 성능분석을 시도하였다. 3차원 SIMD 처리기 구조는 PCI 지역 버스 인터페이스, 16개의 처리기, 그리고 Park's 다중접근기억장치로 구성되며, 다중접근 기억장치는 17개의 외부 메모리 모듈을 갖는다. 기존의 직렬 퐁 음영법을 SIMD 처리기 구조에 수행될 수 있도록 하나의 다면체를 여러 개의 4x4의 정방형 다면체로 나누어서 처리하는 병렬 퐁 음영법으로 수정하였으며, 하나의 정방형 다면체는 다중접근기억장치가 간격이 1인 블록 접근이 가능하기 때문에 16개의 처리기가 동시에 처리할 수 있다. SIMD 처리기 구조에서 수행되는 병렬화된 퐁 음영법을 하드웨어 모의실험 패키지인 CADENCE사의 Verilog-XL로 모의실험을 수행한 결과 5.14배의 속도향상을 보임을 확인하였다.

An Implementation of 3D Graphic Accelerator for Phong Shading

Hyung Lee[†], Younok Park[††] and Jongwon Park[†††]

ABSTRACT

There have been many researches on the 3D graphic accelerator for high speed by needs of CAD/CAM, 3D modeling, virtual reality or medical image. In this paper, an SIMD processor architecture for 3D graphic accelerator is proposed in order to improve the processing time of the 3D graphics, and a parallel Phong shading algorithm is presented to estimate performance of the proposed architecture. The proposed SIMD processor architecture for 3D graphic accelerator consists of PCI local bus interface, 16 Processing Elements (PE's), and Park's multi-access memory system (MAMS) that has 17 memory modules. A serial algorithm for Phong shading is modified for the architecture and the main key is to divide a polygon into 4x4 squares. And, for processing a square, 4 PE's are regarded as a PE Group logically. Since MAMS can support block access type with interval 1, it is possible that 4 PE Groups process a square at a time. In consequence, 16 pixels are processed simultaneously. The proposed SIMD processor architecture is simulated by CADENCE Verilog-XL that is a package for the hardware simulation. With the same simulated result as that of the serial algorithm, the speed enhancement by the parallel algorithm to the serial one is 5.68.

1. 서 론

컴퓨터의 속도 증가와 다양한 그래픽 사용 환경의 급속한 발전으로 인한 고해상도 그래픽 사용 환경의 보편화로 삼차원 그래픽 처리에 의한 그래픽 응용이 날로 확대되어 가고 있다[1]. CAD/CAM, 3D 모델링, 가상현실, 게임, 영화, 의료 분야 등과 같은 영상 매체와 관련된 응용분야는 기존의 2차원 아날로그 NTSC (National Television System Committee) /PAL(Phase Alternation by Line)급 영상처리 기술에서 3차원 디지털 HDTV(High Definition Television)급 영상 처

† 정회원, 충남대학교 컴퓨터공학과
†† 한국전자통신연구원 선임연구원
††† 충남대학교 공과대학 정보통신공학과

리 기술로, 개별 매체로 취급되던 오디오, 비디오, 그래픽스 및 텍스트 등이 복합 매체로 취급되는 등[3], 향상된 고기능화, 고품질화 및 실시간 처리로 진행되어 나아가고 있다. 이러한 현상은 오디오, 비디오, 그래픽과 같은 멀티미디어 데이터 처리기술과 멀티미디어 영상정보 저장, 전송 및 가공을 위한 하드웨어와 소프트웨어의 통합 기술로 지칭되는 차세대 영상정보처리 기술을 절실하게 요구하게 되었다. 이러한 시대적, 기술적 요구사항을 만족시키기 위해서는 그래픽 측면에서 영상 데이터 처리량의 기하급수적인 증가와 더불어 실시간 처리가 요구되어 졌다. 그러나, 특정한 삼차원 그래픽 알고리즘의 경우, 복잡하고 다양한 기법으로 처리되지만 반면에 대규모의 반복 연산이 요구되기 때문에 삼차원 그래픽 처리를 실시간으로 처리하기가 어려운 부분들이 많다.

삼차원 그래픽 처리에 관련된 여러 가지 알고리즘 중에서 일반적으로 퐁 음영법(Phong Shading)[3]은 삼차원 그래픽을 이루는 다면체(Polygon)의 음영과 반사 효과를 광원과 시각의 방향에 따른 각 화소(pixel)별 계산을 수행해야 하는데, 삼차원 그래픽처리 기법 중에서 실시간으로 처리하기 어려운 부분 중의 하나이다. 다양한 삼차원 화면 처리 기법을 실시간으로 처리하기 위한 방법에 관련된 다양한 알고리즘[9] 및 소프트웨어들이 제안되고 있으며, 또한 소프트웨어의 한계성을 극복하기 위하여 하드웨어에 의한 처리 방법[6] 등이 지속적으로 연구 개발되고 있다.

본 논문은 다양한 삼차원 그래픽 알고리즘의 실시간 처리를 위한 하드웨어 구현, 즉, 삼차원 그래픽 가속기에 관한 것으로써 대규모의 다중접근 기억장치[5]와 다수의 연산기로 구성된 SIMD(Single Instruction Multiple Data) 처리기 구조를 제시한다. 또한, 제안한 가속기의 성능을 검증하기 위하여 기존의 퐁 음영법을 토대로 병렬 처리가 가능한 방법을 제시한다. 이는 퐁 음영법이 아직도 널리 적용되고 있으면서 많은 처리 시간을 요구하고, 또한, 다른 그래픽 관련 응용 프로그램에 비하여 제안한 가속기에 이식하기가 수월하기 때문이다.

Verilog HDL(Hardware Description Language)를 사용하여 제안한 가속기를 모델링하고 CADENCE사의 하드웨어 모의실험 패키지인 Verilog-XL을 사용하여 본 논문에서 제안한 퐁 음영법 처리 방법이 제안한 가속기에 의하여 실시간으로 수행됨을 검증하였고, 또한, 3차원 그래픽 처리를 위한 다양한 방법들도 병렬 알고리즘을 적용한 가속기에 의하여 실시간 처리의 가능성을 제시한다.

본 논문의 진행은 다음과 같다. 제 2장에서는 삼차원 그래픽 가속기의 연구 개발 동향을, 제 3장에서는 SIMD 처리기 구조를 갖는 3차원 그래픽 가속기를 제안한다. 그리고, 제안한 가속기의 성능을 검증하기 위해 특정 알고리즘으로 선정된 퐁 음영법의 이론적 배경과 병렬 처리의 필요성에 관하여 제 4장에서 기술한다. 제 5장에서는 퐁 음영법을 병렬 처리하기 위한 방법을 제시하고, 제 6장에서는 제 5장에서 제안한 방법으로 구현한 퐁 음영법을 제안한 가속기에 적용하여 모의실험하고, 모의실험으로부터 얻은 결과를 토대로 성능을 분석하였다. 마지막으로, 제 7장에서는 결론을 내린다.

2. 삼차원 그래픽 가속기의 연구 개발 동향

일반적으로 삼차원 그래픽을 처리하는 프로그램이나 삼차원 그래픽 가속기의 성능은 삼차원 그래픽의 기본 요소인 다면체를 초당 처리할 수 있는 개수로 표시될 수 있는데[1], 삼차원 그래픽 알고리즘 처리는 복잡한 부동 소수점 연산을 각 화소마다 반복적으로 계산되어야 하기 때문에 성능의 증가에는 한계성을 갖고 있다.

대표적으로 Sun Microsystems[7] 사는 이러한 문제를 해결하기 위한 방법으로 4개의 부동 소수점 연산자로 구성된 부동소수점 처리기(Floating Point Unit)가 4개로 구성되어진 그래픽 가속기 구조를 제안하고 있으며(그림 1-a), ATI 사는 그래픽의 연산 부분을 수행하기 위하여 여러 개의 엔진으로 구성된 가속기에 수행할 알고리즘을 여러 단계로 나누어 각각의 엔진에서 할당하여 각 단계들이 순차적으로 처리되는 구조 등이 제시되고 있다. 그리고, 3D Labs 사는 64 비트 hyper-pipelined 구조와 2개의 연산기로 그래픽 데이터를 처리하는 구조 등을 제시하고 있다. 이러한 기존의 가속기들의 공통된 특징은 순차적인 프로그램의 수행을 기본으로 하고, 프로그램의 일부 알고리즘 처리를 복잡하게 구현된 연산 처리기에 분배하여 처리하는 구조를 갖고 있다(그림 1-b). 그러한 방법 외의 추가적인 방법들은 연산의 처리를
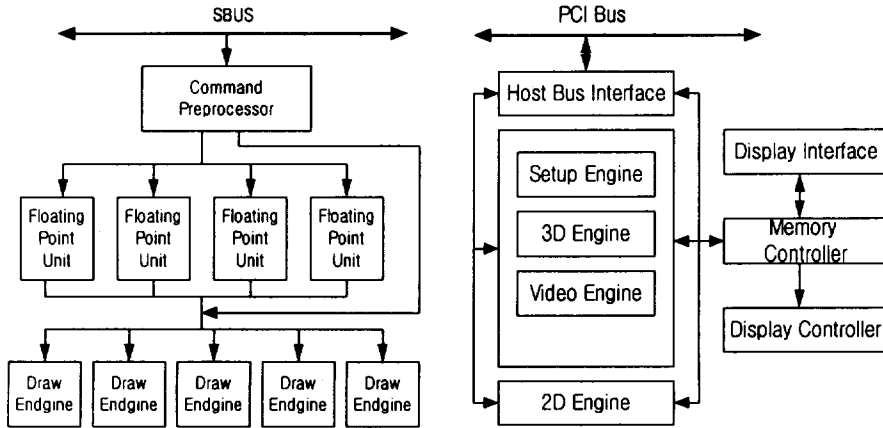
그림 1. (a) Sun Microsystem, (b) ATI에서 제안한 삼차원 그래픽 가속기 구조.

Pipelined 기법[2]을 사용하여 처리 속도를 증대시키는 방법, 가속기 내의 버스 폭을 늘려 1회의 입출력 데이터량을 늘리는 방법[11] 등이 사용되고 있다.

이러한 구조들은 다음과 같은 단점들을 갖고 있는데, 첫째는 병렬화의 단위가 크고, 둘째는 구조상으로 확장성이 없다. 그리고, 영상의 기본 구조인 화소를 직접 사용하지 않고 직렬 수행 방법에 의한 좌표 계산을 매 프로그램 시마다 수행해야 한다. 그래서, 각 화소에 관련된 데이터를 직접 화소별로 처리하지 못하고, 이를 해결하기 위한 좌표 변환이 많은 처리 시간을 요구해서 전반적으로 처리 속도의 향상에 한계가 있다. 이러한 한계성을 극복하기 위해서는 고도의 병렬성, 데이터와 화소간의 일치성을 보장하여 연산 데이터와 영상 데이터 사이의 매핑(mapping) 시간을 줄임으로써 가능하다. 따라서, 앞으로 연구 개발될 삼차원 그래픽 가속기는 파이프라인 기법[2], 병렬화 기법[5], 연산 데이터와 영상 데이터를 동일하게 처리 할 수 있는 기술의 혼합된 구조[1]로 구현될 것이다.

3. 3차원 그래픽 가속기 구조

본 논문에서 제안한 3차원 그래픽 가속기는 SIMD 구조로써 호스트와 데이터 및 명령 전송을 위한 PCI (Peripheral Component Interconnection) 버스, 처리기들, 그리고, 다중접근 기억장치로 구성되었으며, 그림 2와 같다.
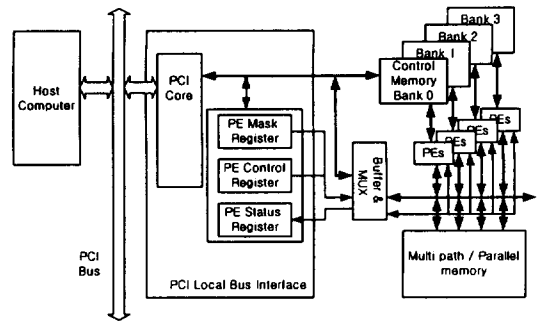
그림 2. 제안한 3차원 그래픽 가속기를 위한 SIMD 처리기 구조

방대한 양의 데이터를 효율적으로 수행하기 위해서 Park's 다중접근 기억장치[4,5]를 가속기의 메모리 구조로써 적용하였는데, 이는 효율적인 메모리 접근 모드를 제공하여 여러 개의 처리기들로부터 출력 데이터를 화면의 1024x764 화소와 1대 1로 대응시키기 위한 별도의 계산 없이 프레임 메모리(frame memory)와 연산 결과를 일치시킬 수 있으며, (r,c)-based 도메인에 적합한 논리적인 2차원 메모리 접근 방식을 제공하기 때문이다.

다중접근 기억장치는 메모리 모듈 선택 회로, 데이터 입출력 라우팅 회로, 주소 계산 및 라우팅 회로, 그리고, 메모리 모듈로 구성되어 있으며(그림 3), 블록, 행, 열 모드의 다양한 간격으로 메모리에 접근할 수 있는 기능을 제공한다. 4x4 정방형 다면체를 처리하기 위해서 다중접근 기억장치를 통해 간격 1의 블

록 모드 방식으로 16개의 처리기들이 동시에 프레임 메모리에 접근할 수 있다.
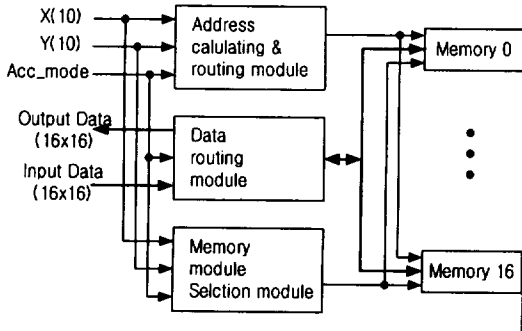
그림 3. 다중접근 기억장치의 블록도

PCI 인터페이스는 128Bytes/s의 전송 속도로 데이터 전송을 수행하며, 처리기들은 알고리즘의 최종 연산부를 수행하도록 설계하였다. 다중접근 기억장치를 16-비트 칼라를 표현할 수 있도록 최대 1024x764 크기의 프레임을 지원하도록 구성하였다. PCI 인터페이스에서 생성되는 어드레스와 데이터 신호는 PCI 인터페이스 내의 제어 및 상태 레지스터와 SIMD 처리기가 공유하고 있으며, PCI 인터페이스 내의 레지스터는 처리기 제어 레지스터와 처리기 상태 레지스터, 그리고, Base Address 레지스터로 구성되어 있다. Base Address레지스터는 PCI 인터페이스 규격의 초기화 과정에서 PCI 버스 타겟 모드로 동작할 경우에는 메모리 맵의 번지를 기입하게 되어 있는데 본 논문에서 설계한 PCI 버스 인터페이스는 Memory mapped I/O 모드로 엑세스하는 것을 기본으로 하였다. 따라서 기입된 어드레스의 상위 8 비트(addr [31:24])의 값을 보드 선택의 비교 값으로 사용하도록 하였다. 실제적인 연산 처리를 담당하는 처리기들은 4개의 처리기 그룹으로 구성되었으며 각각의 처리기 그룹은 4개의 처리기와 제어장치로 구성되었다.

다중접근 기억장치에 저장된 데이터를 한번에 읽어 출력되는 크기를 최대 2 x 16 Bytes가 가능하도록 하였다. 그러나, PCI 버스는 32비트 버스이기 때문에 출력 데이터를 한번에 읽을 수 없고 8번에 걸쳐 나누어서 읽어야 한다. 그래서, PCI 인터페이스의 로컬 데이터 라인과 다중접근 기억장치의 연결은 16 to 2 MUX 기능을 갖는 버퍼로 연결되었다.

처리기는 그래픽에 관련된 복잡한 연산들을 하드와이어(hardwired) 로직으로 수행한다. 따라서 처리기는 복수의 곱셈기, 덧셈기의 조합으로 구성되어 있으며, 연산의 단계별 데이터는 별도의 레지스터를 사용하여 저장한다. 그리고, 제어 메모리는 최초 연산기에 필요한 입력 데이터를 읽어오는 경우에만 접근하고, 연산에 필요한 초기값을 읽어들인 이후에는 접근하지 않는다. 또한, 처리기는 상태 제어기 형태로 구현되며, 연산 속도의 증가를 위해 2개의 산술논리연산장치(ALU : Arithmetic Logic Unit)를 두어 계산 속도를 2배로 늘리도록 구성하였다(그림 4). 상태 제어기는 제어 메모리로부터 입력되는 데이터를 피연산자 레지스터(Operand Register)에 저장하고 데이터의 흐름 및 ALU의 동작 모드를 제어하며, 최종 연산 결과를 다중접근 기억장치로 출력하도록 구성되었다.
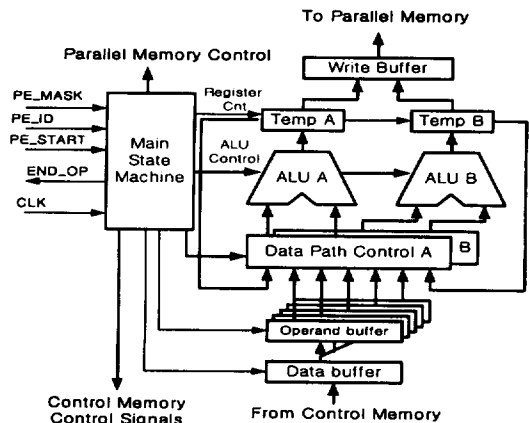
그림 4. 2개의 ALU를 갖는 PE 구조의 블록도.

4. 퐁 음영법(Phong Shading)

퐁 음영법은 주사선 상의 모든 점에서 표면 법선의 근사값을 계산한 후 근사 법선을 사용하여 광도를 계산하는데, 이 방법은 법선 벡터 보간방식(normal-vector interpolation scheme)이라고도 한다. 퐁 음영법은 먼저 주사선 상의 경계점들의 법선 벡터를 보간한다. 그림 5에서 보이는 다각형에서의 점 5의 법선 벡터는 정점 1과 정점 2에서 계산된 법선으로부터 보간되고, 점 6의 법선은 정점 1과 정점 4의 법선으로부터 계산된다. 주사선(scan line) 상의 모든 내부점들은 정점 5와 정점 6의 법선으로부터 보간된 법선이

부여된다. 이렇게 함으로써 정점들로부터 직접 보간한 광도보다 더 정확한 결과를 얻을 수 있다.
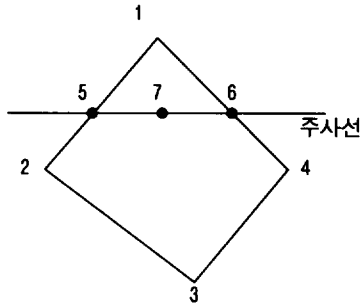
그림 5. 다각형을 지나는 주사선 상의 내부점 7의 법선은 정점 5와 6의 법선으로부터 보간 된다.

대부분의 경우 실시간으로 음영처리를 수행하기 위해서 구로우 음영법을 이용하지만, 표면의 하이라이트가 때로는 비정상적인 형태로 비추어지며 선형 광도 보간법 때문에 마하띠(Mach Band)라고 하는 밝거나 어두운 광도의 띠가 표면에 있을 수 있다. 퐁 음영법은 이러한 flattening out과 dull surface를 제거하고 마하띠를 축소시켜 고화질의 하이라이트를 표현할 수 있기 때문에 현실감을 살릴 수 있지만, 일반적인 퐁 음영법은 각 화소마다 3번의 덧셈, 1번의 나눗셈, 그리고 1번의 평방근을 필요로 하기 때문에 실시간 처리가 어렵다[10]. 이러한 많은 계산량을 줄이기 위해서 제시된 빠른 퐁 음영법(Fast Phong Shading)은 각 화소마다 램버트 반사(Lambertian reflection)을 위한 2번의 덧셈과 그 외에 5번의 덧셈과 1번의 메모리 접근(memory access)을 필요로 한다[10]. 그렇지만, 기계어 레벨에서 수행되는 연산을 분석해 보면 일반적인 퐁 음영법[15]을 수행하기 위해서는 하나의 화소마다 23개의 부동 소수점 곱셈, 4개의 부동 소수점 나눗셈, 7개의 부동 소수점 덧셈, 그리고, 1개의 제곱근 연산을 수행해야 한다.

5. 퐁 음영법을 고속 처리하기 위한 병렬화 방법

대부분의 삼차원 그래픽 처리는 각각의 다면체를 처리하는데, 퐁 음영법에서도 각각의 다면체를 하나의 단위로 취급하여 음영처리 한다. 다면체의 크기와 모양은 일정하지 않으나 다각형눈 그물 표시법(Polygon-mesh representation)으로 표현된 대부분의 모델은 삼각형 또는 사각형 모양의 다면체들로 구성된다. 사각형 모양의 다면체들로 표현된 모델의 음영을 나타내기 위하여 퐁 음영법을 적용할 경우, 화소에 따른 법선 벡터를 계산하기 위한 일련의 연산들이 다면체 내의 화소 수만큼 반복적으로 처리되어야 한다. 주사선 상의 모든 화소의 법선 벡터를 계산함으로써 다면체 내부의 화소들에 대한 법선 벡터를 구할 수 있는데, 이러한 방법에서는 다면체들의 형태가 다양하기 때문에 한 정점을 교차점으로 서로 다른 두 정점을 잇는 두개의 직선과 주사선이 교차하는 점들의 개수가 일정하지 않은 경우가 빈번하게 발생한다. 즉, 다면체를 이루는 임의의 서로 다른 정점 1, 2, 3으로부터 정점 1과 정점 2를 잇는 직선상의 점들을 집합 A, 정점 1과 정점 3을 잇는 직선상의 점들을 집합 B, 그리고, 주사선이 $x \in A$이고 $y \in B$이며 $x \neq y$인 단사함수 (injective function) $f(x)=y$로 표현되어진다면 모든 x에 대하여 함수 F를 만족하는 어떤 y의 집합 B'과 집합 B 사이에 $B' \subset B$인 관계가 성립된다. 병렬화의 단위가 다면체보다 작은 크기로 나누어질 경우, $y \in B-B'$를 만족하기 위한 $x' \not\in A$인 x'를 계산하기 위한 오버헤드(overhead)가 발생하기 때문에 일관성 있는 병렬화 처리에는 부적합하다. 그래서, 일반적인 가속기에서는 다면체를 하나의 처리 단위로 간주하고, 다면체을 형성하는 각 정점들의 법선 벡터를 가속기에 전송하여 처리하도록 한다. 처리된 데이터는 다시 각 화소에 따른 좌표를 다시 계산해야 한다. 이렇게 계산된 좌표와 일치하는 비디오 메모리에 화소의 법선 벡터를 저장함으로써 모든 과정이 완료된다.
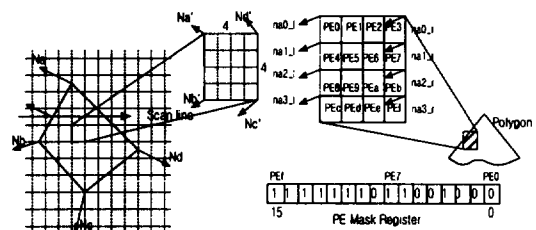
그림 6. (a)다면체를 4x4의 정방형 다면체로 분할 (b) 정방형 다면체와 마스크 레지스터

본 논문에서는 앞서 언급한 문제들을 해결할 수 있는 병렬처리 방법을 제시한다. 병렬화를 위해서는 다면체가 일정한 크기와 규칙적인 배열성을 갖고 있어야 한다[4]. 이러한 규칙성, 배열성, 일정한 크기를

부여하기 위하여 본 논문에서는 주어진 다면체를 다시 4x4개의 화소을 갖는 정방형의 다면체로 분해하는 방법(그림 6-a)을 적용한다. 이러한 방법을 적용하기 위해서, 호스트 컴퓨터는 처리해야 할 데이터를 48x1의 크기 단위로 법선 벡터를 계산한다. 예를 들면, 좌표 (x, y)과 (x+47, y)의 법선 벡터를 계산한다. 이러한 연산을 반복 수행하여 48 x N개의 다면체로 재구성하여 제안한 가속기에 전송한다. 제안한 가속기 내의 16개의 처리기(PE: Processing Element)가 간격이 3인 행 접근으로 데이터를 읽어 온다. 이때, PE_0와 PE_f만이 호스트 컴퓨터에서 계산된 유효한 법선 벡터를 읽어 오게 된다. 읽어 온 2개의 법선 벡터를 토대로 PE_1과 PE_e는 동일한 법선 벡터 계산을 수행하여 간격이 3인 행 접근으로 데이터를 저장한다. 이렇게 4번에 걸쳐서 수행하게 되면 4x4를 수행하기 위한 정방형 다각형이 구성되고, 간격이 1인 블록 접근을 수행하여 나머지 화소들의 법선 벡터를 계산한다. 이러한 수행은 처리기 그룹이 4개의 처리기와 1개의 제어장치로 구성되어 있으며, 각각의 처리기는 처리기 그룹 내에서 몇 번째 처리기인지, 16개의 처리기 내에서 몇 번째 처리기인지를 알고 있다. 예를 들면, 두 번째 처리기 그룹 내의 두 번째 처리기는 그룹 내에서는 두 번째, 전체 16개의 처리기 중에서 여섯 번째임을 알고 있다. 이는 퐁 음영법을 수행하기 전에, 초기화 단계에서 처리기 내의 레지스터에 저장하기 때문이다. 이러한 병렬처리를 수행하기 위한 퐁 음영법의 흐름도는 그림 7과 같다.
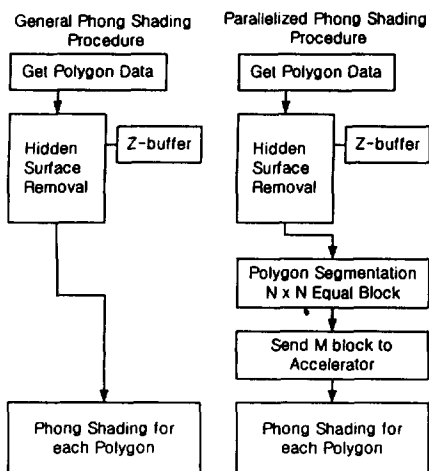
그림 7. 기존의 퐁 음영법과 제안한 퐁 음영법의 병렬 방법

음영법을 적용할 물체의 형태가 다양하고 제안한 가속기가 48 x N 형태를 전송 받아서 처리하기 때문에 물체를 포함하는 최소 다각형을 이루도록 하였으나 주사선 상에서 물체의 모서리가 48의 배수가 아닐 수 있다. 이런 경우, 가속기는 하나의 법선 벡터(PE_0에 한정된)를 읽어오거나 또는 어떠한 법선 벡터도 읽어오지 못한다. 이러한 문제를 해결하기 위해서 호스트 컴퓨터는 주사선 상의 특정 내부점의 법선 벡터를 계산한다. 즉, 모서리에 가장 근접하면서 시작점 x를 기준으로 x+3n인 위치에 존재하는 내부점의 법선 벡터를 계산한다. 이렇게 함으로써 특정 내부점들은 제안한 가속기에서 특정 PE에 읽혀지게 되며, 제어장치가 마스크 비트를 체크하여 어떤 PE가 호스트 컴퓨터에서 계산된 법선 벡터를 읽어 왔는지를 알 수 있다. 이 때, 간격이 3인 행 접근으로 처리기들이 데이터를 읽었기 때문에 주사선 상의 모서리에 대한 법선 벡터는 가속기의 메모리 내부에 저장되어 있지만 처리기에 읽혀지지는 않는다. 이와 같이 간격이 3인 행 접근으로 데이터를 읽어 들이고 계산을 수행한 다음, 수행한 결과를 간격이 3인 행 접근으로 저장하는데, Y 좌표를 증가시켜 가면서 4회 반복 수행하게 되면 가속기의 메모리 내부의 데이터는 4x4 정방형 다면체로 재구성되게 된다. 그래서, 간격이 1인 블록 접근을 통해서 4x4내의 16개의 데이터를 읽게 되는데, 첫 번째 처리기 그룹의 경우에는 PE_0와 PE_3이 이전에 계산된 법선 벡터를 읽어 왔으므로 나머지 PE_1과 PE_2의 법선 벡터를 계산할 수 있다. 이 때, 고려해야 할 문제는 모서리를 포함할 경우이다. 이 경우도 앞선 경우와 마찬가지로 마스크 비트를 두어서 해결하였다. 즉, 그림 6-(b)와 같이 4x4 정방형 다면체 내에 존재하는 사선 밖의 화소에 대해서는 처리기로 하여금 연산을 수행하지 않도록 하기 위해 마스크 비트(mask bits)를 두었으며, 마스크 비트의 값이 '0'인 경우에는 해당 처리기가 연산을 수행하지 않도록 하였다.

6. 모의 실험 및 성능 분석

모의 실험은 Verilog HDL(Hardware Description Language)을 사용하여 SIMD 처리기를 기술하고 CADENCE사의 모의실험 패키지인 Verlog-XL을 사용하여 모의실험을 수행하였다. 그림 8에서 A_D
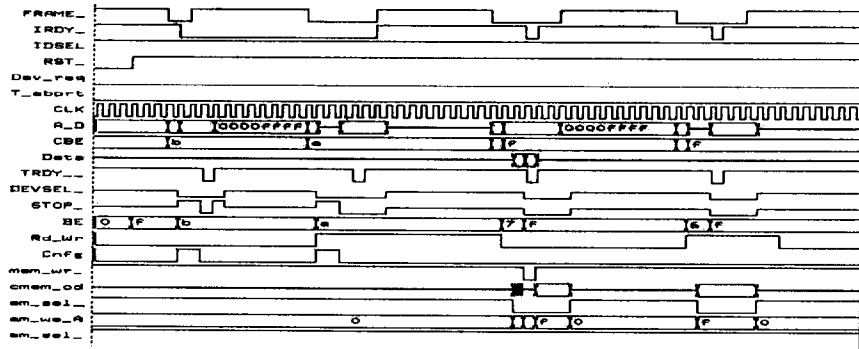
그림 8. PCI Local Bus Interface에 의한 제어 메모리 데이터 입출력 파형.

는 PCI 버스상의 어드레스 데이터 멀티플렉스된 신호이며, c_mem_od는 제어 메모리(control memory)의 출력 데이터 이다. CLK은 33MHz의 PCI 버스 주기이며, TRDY_, FRAME_, C_BE_, IRDY_, DEVSEL_, STOP_, T_abort, IDSEL 등은 PCI 규격에 의한 PCI 버스 인터페이스 신호선 들이다. mem_wr_ 신호는 SIMD 처리기 내의 모든 레지스터, 제어 메모리, 다중접근 기억장치에 공통적으로 적용되는 신호로써 memory write 시점을 알려주기 위한 시점이며, 이 신호와 어드레스를 디코딩하여 접근하고자 하는 모듈의 write enable 신호로 사용한다. FRAME_ 신호는 PCI 버스 전체를 총괄하는 신호로써 버스 구동의 시작과 끝을 알려주는 신호이다.

설계된 회로의 정확한 성능 측정값을 얻기 위해서는 일반적으로 다음과 같은 과정을 거쳐야 한다. 설계한 회로를 HDL로 기술하고, 모의실험을 통한 기능을 검정을 거친 다음, ASIC vendor가 제공하는 라이브러리로 합성을 수행한다. 그리고, 라이브러리에서 제공되는 셀(cell)의 지연 시간(delay time)을 계산하여 모의실험을 수행한 후 실제 로직이 적재될 ASIC vendor가 제공하는 Back-end Tool을 사용하여 칩의 다이(die)에서의 layout이 이루어진 상태에서의 routing delay와 fanout delay를 포함한 지연 시간을 계산하여 모의실험을 수행해야 하는 Back-end 과정을 거쳐야 한다. 그러나, 본 논문에서 제안한 삼차원 그래픽 가속기는 Verilog-XL를 사용하여 기능 검증을 위한 모의실험을 수행하여 하드웨어 구현 시 예상되는 속도를 측정하였는데, 이는 Front-End 과정만을 거친 단순 지연시간(delay time)을 사용하여 성능을 예측한 것이다.

예측된 성능을 바탕으로 1개의 다면체(64 pixels) 처리에 대하여 120MHz에서 동작하는 Pentium Processor를 장착한 PC와의 성능을 비교 분석하였다. 즉, 제어 메모리에 4x4 화소로 정규화된 다면체로 형성된 데이터가 SIMD 처리기로 전달되어 PE에서 수행한 후 다중접근 기억장치에 저장될 때까지의 수행 시간($T_{parallel}$)과 PC에서 직렬 프로그램을 수행하여 디스플레이 장치로 데이터를 넘겨주기까지 걸리는 수행 시간(T_{serial})을 1개의 다면체를 바탕으로 비교하였다.

$$T_{serial} = \sum^{32}\left(\sum^{12} T_{PC_ADD} + \sum^{20} T_{PC_MUL} + \sum^{5} T_{PC_DIV} + T_{MEM}\right) + \alpha,$$

$$T_{parallel} = T_{PCI} \times 8 + \sum^{4}\left(\sum^{12} T_{ADD} + \sum^{20} T_{MUL} + \sum^{5} T_{DIV} + T_{MAMS}\right) + \beta,$$

여기서, T_{PC_ADD}=10*ns*, T_{PC_MUL}=30*ns*, T_{PC_DIV}=45*ns*, T_{MEM}=70*ns*인데, 이들 값은 120MHz에서 동작하는 PC로부터 얻은 값이고, T_{ADD}=2*ns*, T_{MUL}=12*ns*, T_{DIV}=32*ns*, T_{MAMS}=210, T_{PCI}=33*ns*는 SIMD처리기를 ASIC으로 구현했을 경우를 가정하여 얻은 값이며, α와 β는 연산기 처리 시간 이외의 기타 지연 시간들을 고려한 변수들인데, 특히, β는 구현상에서 발생할 수 있는 제어를 위한 동기 확립을 위한 지연 시간 등에 관한 변수이다. 만약, $\alpha \cong \beta$라고 가정할 경우, T_{serial}는 32,480*ns*, $T_{parallel}$는 2,800*ns*이며, SIMD처리기가 11.6배의 속도 향상을 보였다. 만약, 덧셈기, 곱셈기, 그리고, 나눗셈기가 SIMD처리기를 구현한 ASIC 소자와 같은 처리속도를 갖는다고 한다면 T_{serial}=14,408*ns*이며, 이런 경우에서는 SIMD처리기가 5.14배의 속도 향상을 보인다.

7. 결 론

본 논문에서는 SIMD 처리기에 적합하고 병렬화가 가능한 퐁 음영법을 제안하였고, 퐁 음영법을 처리하는 부분 중 반복적이면서도 연산량이 많은 부분을 SIMD 처리기에서 병렬로 수행함으로써 속도면에서 성능이 상당히 향상되었음을 볼 수 있었다. 그리고, 병렬화된 부분의 수행을 위해 SIMD 처리기를 개발하였고, 직접 모의실험을 함으로써 병렬 수행이 가능함과 동시에 하드웨어로 구현 할 수 있음을 보였다. 또한, 제안된 SIMD 처리기 내의 처리기그룹의 개수를 늘림으로써 더 많은 용량의 다면체 처리가 가능하며, 처리 속도를 더욱 향상시키기 위한 처리기의 확장성을 충분히 고려할 수 있었다.

본 논문에서는 처리기가 처리하는 수행부를 퐁 음영법으로 제한하였지만, 삼차원 그래픽에서 적용되는 대부분의 알고리즘이 다면체 단위로 이루어지므로 임의의 다면체를 정방형 다면체로 나누어 병렬처리하는 방법의 적용이 가능하다. 따라서, 다양한 삼차원 그래픽 알고리즘을 지원하는 삼차원 그래픽 가속기를 구현하기 위해서는 본 논문에서 제시한 병렬 처리 방법을 적용하고, 처리기가 각 알고리즘에서 공통적인 연산 부분과 수행과정을 모두 수용할 수 있도록 설계되면 각 처리기가 화소단위 처리를 수행할 수 있다. 이 경우, 처리기의 효율을 극대화하기 위해서 다중접근 기억장치가 제공하는 다양한 간격을 허용하는 접근 모드 방식으로 동시에 다면체를 엑세스 할 수 있어야 한다.

앞으로의 과제는 제안한 병렬화 방법을 범용으로 적용하기 위한 처리기의 개량과 PCI 인터페이스 통과에 따른 전송 속도 저하를 방지하기 위하여 다중접근 기억장치와 비디오 메모리를 일체화 할 수 있는 구조의 연구 등이 진행되어야 할 것이다.

참 고 문 헌

[1] H.K. Reghbati and A.Y.C. Lee, "Computer Graphics Hardware: Image Generation and Display," Computer Society Press, 1988

[2] A. Fujimoto, G. P. Christopher, and K. Iwata, "A 3-D Graphics Display System with depth buffer and pipelined processor," IEEE Computer Graphics and Applications, pp. 11-23, June 1984

[3] J. D. Foley, A. Van Dam, S. K. Feiner, and J. F. Huges, Computer Graphics : Principles and Practice, 2^{nd} Edition, Addison-Wesley Publishing Co., 1996

[4] J. W. Park, "An Efficient Memory System for Image Processing," IEEE Trans. Computers, Vol. C-35, No. 7, pp. 169-174, July 1986

[5] 박 종원, "영상처리를 위한 다중접근 기억구조," 한국과학기술원 박사학위논문, 1991

[6] B. MacIntyre, "PC 3D Graphic Accelerator FAQ," URL: http://www.cs.colombia.edu/~bm/3dcards/3d-cardsl.html

[7] J. Clark, "The Geometry engine : A VLSI geometry system for graphics," Proc. NATO Advanced Study Institute on Microarchitecture of VLSI Computers, pp. 189-205, 1985

[8] W. Myers, "Staking out the graphics display pipeline," IEEE Computer Graphics and Applications, pp. 60-65, July 1984

[9] N. England, "A graphics system architecture for interactive application-specific display functions," IEEE Computer Graphics and Applications, pp. 60-70, Jan. 1986

[10] Gray Bishop and David M. Weimer, "Fast Phong Shading," SIGGRAPH'86 Proceedings, Vol.20, pp. 103-106, Aug. 1986

[11] Dipak Ghosal and L. M. Patnaik, "Parallel Polygon Scan Conversion Algorithm : performance Evaluation on a shared bus architecture," Computer and Graphics, Vol. 10, No. 1, pp. 7-25, 1986

[12] Arie Kaufman, "Memory Organization for a Cubic Frame Buffer," Eurographics '86, pp. 93-100, Aug. 1986

[13] Frederik W. Jansen, "CSG Hidden Surface Algorithm for VLSI Hardware Systems," Advances in Computer Graphics Hardware I, Record of First Eurographics Workshop on Graphics Hardware, pp. 75-82, 1986

[14] William H. Press and Brain P. Flannery,

"Numerical Recipes in C," Cambridge University Press, 1988

[15] Tom Duff, "Smoothly Shaded Renderings of Polyhedral Objects on Raster Displays," ACM Computer Graphics, Vol. 13, No. 2, pp. 270-275, 1979

이 형

1995년 2월 충남대학교 컴퓨터과학과 졸업, 이학사
1997년 2월 충남대학교 컴퓨터공학과 졸업, 공학석사
2000년 2월 충남대학교 컴퓨터공학과 박사과정 수료
관심분야 : 영상처리, 병렬처리, 반도체 설계

박 윤 옥

1986년 2월 한양대학교 전자공학과 졸업, 공학사
1985년 12월~1986년 1월 삼성전자 종합연구소 연구원
1986년 2월 현재 한국전자통신연구원 선임연구원
관심분야 : 이동 통신 변복조 모뎀설계, 영상 신호 처리

박 종 원

1979년 2월 충남대학교 전자공학과 졸업, 공학사
1981년 2월 한국과학기술원 전산학과 졸업, 전산학 석사
1991년 8월 한국과학기술원 전산학과 졸업, 전산학 박사
1995년~현재 충남대학교 공과대학 정보통신공학과 정교수
관심분야 : 영상처리, 병렬처리, 의공학